Research Article
Computing Simple Roots by an Optimal Sixteenth-Order Class

F. Soleymani, 1 S. Shateyi, 2 and H. Salmani 3

1 Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran
2 Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa
3 Department of Civil Engineering, Islamic Azad University, Zahedan Branch, Zahedan, Iran

Correspondence should be addressed to S. Shateyi, stanford.shateyi@univen.ac.za

Received 21 August 2012; Revised 6 October 2012; Accepted 7 October 2012

Academic Editor: Changbum Chun

Copyright © 2012 F. Soleymani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem considered in this paper is to approximate the simple zeros of the function \( f(x) \) by iterative processes. An optimal 16th order class is constructed. The class is built by considering any of the optimal three-step derivative-involved methods in the first three steps of a four-step cycle in which the first derivative of the function at the fourth step is estimated by a combination of already known values. Per iteration, each method of the class reaches the efficiency index \( \sqrt[16]{5} \approx 1.741 \), by carrying out four evaluations of the function and one evaluation of the first derivative. The error equation for one technique of the class is furnished analytically. Some methods of the class are tested by challenging the existing high-order methods. The interval Newton’s method is given as a tool for extracting enough accurate initial approximations to start such high-order methods. The obtained numerical results show that the derived methods are accurate and efficient.

1. Introduction

Consider that the function \( f : I \subseteq \mathbb{R} \rightarrow \mathbb{R} \) is a sufficiently differentiable scalar function. We assume that \( r \in I \) be a simple zero of the nonlinear function, that is, \( f(r) = 0 \) and \( f'(r) \neq 0 \). There is a vast literature on simple zeros of such nonlinear functions by iterative processes, see, for example, [1–3]. In 1974, Kung and Traub [4] conjectured that a multipoint iterative method without memory for solving single variable nonlinear equations consisting of \( n + 1 \) evaluations per iteration has the maximal convergence rate \( 2^n \) and, subsequently, the maximal efficiency index will be \( \sqrt[2^n]{n} \). By taking into consideration this concept, many authors, see, for example, [5–7], have tried to produce optimal multistep methods.
Kung and Traub in [4] presented the following $n$-point optimal iterative process consisting of $(n + 1)$-evaluation per full cycle:

$$
q_j(f)(x_n) = x_n - \frac{f(x_n)}{f'(x_n)},
$$

(1.1)

$$
\vdots
$$

$$
q_{j+1}(f)(x_n) = S_j(0),
$$

for $j = 2, \ldots, n - 1$, where $S_j(y)$ is the inverse Hermite interpolatory polynomial of degree at most $j$ such that

$$
S_j(f(x_n)) = x_n, \quad S'_j(f(x_n)) = \frac{1}{f'(x_n)},
$$

(1.2)

$$
S_j(f(q_\lambda(f)(x_n))) = q_\lambda(x_n), \quad \lambda = 2, \ldots, j.
$$

This technique is defined by $x_{n+1} = q_n(f)(x_n)$ starting with an initial point $x_0$. Recently, Neta and Petković applied the concept of inverse interpolation again in [8] to approximate the first derivative of the function in the third and fourth steps of a three- and four-step cycle. They obtained the following optimal 8th order technique:

$$
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
$$

$$
z_n = y_n - \frac{f(x_n) + tf(y_n)}{f(x_n) + (t-2)f(y_n)} \frac{f(y_n)}{f'(x_n)},
$$

(1.3)

$$
x_{n+1} = y_n + c[f(x_n)]^2 - d[f(x_n)]^3,
$$

with $t \in \mathbb{R}$, and $c, d$ as comes next

$$
d = \frac{1}{(f(y_n) - f(x_n))(f(y_n) - f(z_n)) f[y_n, x_n]} - \frac{1}{(f(z_n) - f(x_n))(f(y_n) - f(z_n)) f[z_n, x_n]},
$$

$$
+ \frac{1}{f'(x_n)(f(z_n) - f(x_n))(f(y_n) - f(z_n))} - \frac{1}{f'(x_n)(f(y_n) - f(x_n))(f(y_n) - f(z_n))},
$$

(1.4)

$$
c = \frac{1}{(f(y_n) - f(x_n)) f[y_n, x_n]} - \frac{1}{f'(x_n)(f(y_n) - f(x_n))} - d(f(y_n) - f(x_n)).
They also presented an optimal 16th order technique in the following structure consisting of four evaluations of the function and one evaluation of the first derivative:

\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]

\[
z_n = y_n - \frac{f(x_n) + tf(y_n) f(y_n)}{f(x_n) + (t-2)f(y_n) f'(x_n)},
\]

\[
w_n = y_n + c[f(x_n)]^2 - d[f(x_n)]^3,
\]

\[
x_{n+1} = y_n + c'[f(x_n)]^2 - d'[f(x_n)]^3 + g'[f(x_n)]^4,
\]

wherein (without the index \(n\))

\[
g' = \frac{(\varphi_w - \varphi_z)(f(w) - f(z)) - ((\varphi_y - \varphi_z)(f(y) - f(z)))}{f(w) - f(y)},
\]

\[
d' = \frac{\varphi_w - \varphi_z}{f(w) - f(z)} - g'(f(w) - 2f(x) + f(z)),
\]

\[
c' = \varphi_w - d'(f(w) - f(x)) - g'(f(w) - f(x))^2,
\]

and also \(\varphi_w = 1/(f\{w,x\}(f(w) - f(x))) - 1/(f'(x_n)(f(w) - f(x))), \varphi_z = 1/(f\{z,x\}(f(z) - f(x))) - 1/(f'(x_n)(f(z) - f(x))), \) and \(\varphi_y = 1/(f\{y,x\}(f(y) - f(x))) - 1/(f'(x_n)(f(y) - f(x))).\)

For further reading, one may refer to [9, 10]. Contents of the paper are summarized in what follows. In the next section, our novel contribution is constructed by considering an optimal eighth-order method in the first three steps of a four-step cycle, in which the derivative in the quotient of the new Newton’s iteration is estimated such that the order remains at 16, namely, optimal according to the Kung-Traub hypothesis. The new class of methods is supported with detailed proof in this section to verify the construction theoretically.

Then, it will be observed that the computed results listed in Table 2 completely support the theory of convergence and efficiency analyses discussed in the paper. Section 4 reminds the well-known Interval Newton method in the efficient programming package Mathematica 8 to extract enough accurate initial guesses to start the process. Finally, a short conclusion is given in the last section.

2. Main Results

In this section, we derive our main results by providing a class of four-step iterative methods, which agrees with Kung-Traub hypothesis. In order to build the class, we consider the
following four-step four-point iteration, in which the first three steps are any of the optimal three-step three-point without memory derivative-involved methods:

\[
\text{optimal 8th-order method} = \begin{cases} 
  f(x_n) \text{ and } f'(x_n) \text{ are available}, \\
  f(y_n) \text{ is available}, \\
  f(z_n) \text{ is available}, 
\end{cases}
\]

\[ x_{n+1} = w_n - \frac{f(w_n)}{f'(w_n)}. \]  

(2.1)

It can be seen that per full cycle of the structure (2.1), it includes four evaluations of the function and two evaluations of the first derivative to reach the convergence rate 16. At this time, we should approximate the first derivative of the function at the fourth step such that the convergence order does not decrease. In order to do this effectively, we have to use all of the known data from the past steps, that is, \( f(x_n), f'(x_n), f(y_n), f(z_n) \), and \( f(w_n) \). Herein, we take into account the nonlinear fraction given by (without the index \( n \))

\[ p(t) = \frac{b_1 + b_2(t-x) + b_3(t-x)^2 + b_4(t-x)^3}{1 + b_5(t-x)}. \]  

(2.2)

This nonlinear fraction is inspired by Pade approximant in essence. The approximation function should meet the interpolation conditions \( f(x_n) = p(x_n), f'(x_n) = p'(x_n), f(y_n) = p(y_n), f(z_n) = p(z_n) \), and \( f(w_n) = p(w_n) \). Note that the first derivative of (2.2) takes the following form:

\[ p'(t) = \frac{b_2 - b_1 b_5 + 2b_3(t-x_n) + (3b_4 + b_2 b_5)(t-x_n)^2 + 2b_4 b_5(t-x_n)^3}{(1 + b_5(t-x_n))^2}. \]  

(2.3)
Table 2: Results of comparisons for different methods.

<table>
<thead>
<tr>
<th>Functions and $x_0$</th>
<th>G-K 16</th>
<th>N-P 16</th>
<th>S-S 16 (I)</th>
<th>S-S 16 (II)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$, $x_0 = -1.2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>0.5e - 12</td>
<td>0.9e - 15</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>0.5e - 187</td>
<td>0.8e - 232</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>0.2e - 2989</td>
<td>0.1e - 3704</td>
</tr>
<tr>
<td>$f_1$, $x_0 = -3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>0.6e - 6</td>
<td>0.6e - 6</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>0.4e - 93</td>
<td>0.1e - 92</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>0.3e - 1487</td>
<td>0.1e - 1427</td>
</tr>
<tr>
<td>$f_2$, $x_0 = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_2(x_1)</td>
<td>$</td>
<td>0.4e - 18</td>
<td>0.3e - 19</td>
</tr>
<tr>
<td>$</td>
<td>f_2(x_2)</td>
<td>$</td>
<td>0.6e - 310</td>
<td>0.4e - 327</td>
</tr>
<tr>
<td>$</td>
<td>f_2(x_3)</td>
<td>$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_2$, $x_0 = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_2(x_1)</td>
<td>$</td>
<td>0.2e - 11</td>
<td>0.7e - 11</td>
</tr>
<tr>
<td>$</td>
<td>f_2(x_2)</td>
<td>$</td>
<td>0.1e - 202</td>
<td>0.5e - 193</td>
</tr>
<tr>
<td>$</td>
<td>f_2(x_3)</td>
<td>$</td>
<td>0.7e - 3260</td>
<td>0.4e - 3107</td>
</tr>
<tr>
<td>$f_3$, $x_0 = 1.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_3(x_1)</td>
<td>$</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>$</td>
<td>f_3(x_2)</td>
<td>$</td>
<td>0.1e - 15</td>
<td>0.1e - 10</td>
</tr>
<tr>
<td>$</td>
<td>f_3(x_3)</td>
<td>$</td>
<td>0.8e - 262</td>
<td>0.2e - 177</td>
</tr>
<tr>
<td>$f_3$, $x_0 = 0.9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_3(x_1)</td>
<td>$</td>
<td>F</td>
<td>0.1e - 4</td>
</tr>
<tr>
<td>$</td>
<td>f_3(x_2)</td>
<td>$</td>
<td>F</td>
<td>0.6e - 109</td>
</tr>
<tr>
<td>$</td>
<td>f_3(x_3)</td>
<td>$</td>
<td>F</td>
<td>0.1e - 2298</td>
</tr>
<tr>
<td>$f_4$, $x_0 = 0.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_4(x_1)</td>
<td>$</td>
<td>0.4e - 8</td>
<td>0.2e - 8</td>
</tr>
<tr>
<td>$</td>
<td>f_4(x_2)</td>
<td>$</td>
<td>0.1e - 129</td>
<td>0.4e - 133</td>
</tr>
<tr>
<td>$</td>
<td>f_4(x_3)</td>
<td>$</td>
<td>0.4e - 2073</td>
<td>0.1e - 2129</td>
</tr>
<tr>
<td>$f_5$, $x_0 = 1.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_1)</td>
<td>$</td>
<td>0.5e - 11</td>
<td>0.1e - 12</td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_2)</td>
<td>$</td>
<td>0.3e - 175</td>
<td>0.1e - 201</td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_3)</td>
<td>$</td>
<td>0.8e - 2803</td>
<td>0.4e - 3225</td>
</tr>
<tr>
<td>$f_5$, $x_0 = 1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_1)</td>
<td>$</td>
<td>0.1e - 8</td>
<td>0.1e - 6</td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_2)</td>
<td>$</td>
<td>0.1e - 139</td>
<td>0.1e - 111</td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_3)</td>
<td>$</td>
<td>0.9e - 2238</td>
<td>0.6e - 1787</td>
</tr>
<tr>
<td>$f_5$, $x_0 = 4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_1)</td>
<td>$</td>
<td>2.42</td>
<td>0.6e - 2</td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_2)</td>
<td>$</td>
<td>0.61</td>
<td>0.1e - 43</td>
</tr>
<tr>
<td>$</td>
<td>f_5(x_3)</td>
<td>$</td>
<td>0.2e - 14</td>
<td>0.4e - 712</td>
</tr>
<tr>
<td>$f_6$, $x_0 = -2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_6(x_1)</td>
<td>$</td>
<td>4.2</td>
<td>0.6</td>
</tr>
<tr>
<td>$</td>
<td>f_6(x_2)</td>
<td>$</td>
<td>0.1e - 8</td>
<td>0.3e - 22</td>
</tr>
<tr>
<td>$</td>
<td>f_6(x_3)</td>
<td>$</td>
<td>0.5e - 158</td>
<td>0.2e - 377</td>
</tr>
<tr>
<td>$f_6$, $x_0 = -0.6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_6(x_1)</td>
<td>$</td>
<td>F</td>
<td>0.5e - 1</td>
</tr>
<tr>
<td>$</td>
<td>f_6(x_2)</td>
<td>$</td>
<td>F</td>
<td>0.2e - 38</td>
</tr>
</tbody>
</table>
Substituting the known data in (2.2) and (2.3) will result in obtaining the five unknown parameters. It is obvious that $b_1 = f(x_n)$, hence we obtain the following system of four linear equations with four unknowns:

$$b_2 - f(x_n) = f'(x_n),$$

$$(y_n - x_n)b_2 + (y_n - x_n)^2b_3 + (y_n - x_n)^3b_4 - (y_n - x_n)f(y_n) = f(y_n) - f(x_n),$$

$$(z_n - x_n)b_2 + (z_n - x_n)^2b_3 + (z_n - x_n)^3b_4 - (z_n - x_n)f(z_n) = f(z_n) - f(x_n),$$

$$(w_n - x_n)b_2 + (w_n - x_n)^2b_3 + (w_n - x_n)^3b_4 - (w_n - x_n)f(w_n) = f(w_n) - f(x_n).$$

**Table 2:** Continued.

<table>
<thead>
<tr>
<th>Functions and $x_0$</th>
<th>G-K 16</th>
<th>N-P 16</th>
<th>S-S 16 (I)</th>
<th>S-S 16 (II)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>f_0(x_0)</td>
<td>$</td>
<td>F.</td>
<td>$0.6e - 636$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 0.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>$0.3e - 8$</td>
<td>$0.4e - 8$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>$0.3e - 142$</td>
<td>$0.9e - 141$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>$0.7e - 2288$</td>
<td>$0.2e - 2264$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>F.</td>
<td>$1.4$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>F.</td>
<td>$0.4e - 8$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>F.</td>
<td>$0.2e - 141$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 1.1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>$0.4e - 9$</td>
<td>$0.1e - 9$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>$0.1e - 162$</td>
<td>$0.4e - 174$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>$0.1e - 2621$</td>
<td>$0.5e - 2806$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 2.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>$0.4e - 1$</td>
<td>$0.1e - 1$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>$0.9e - 36$</td>
<td>$0.6e - 76$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>$0.1e - 590$</td>
<td>$0.5e - 755$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>$0.1e - 19$</td>
<td>$0.2e - 18$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>$0.6e - 335$</td>
<td>$0.1e - 312$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 2.2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>F.</td>
<td>$0.1$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>F.</td>
<td>$0.3e - 21$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>F.</td>
<td>$0.4e - 349$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 0.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>F.</td>
<td>$5.3$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>F.</td>
<td>$0.1e - 13$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>F.</td>
<td>$0.6e - 246$</td>
</tr>
<tr>
<td>$f_1$, $x_0 = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_1)</td>
<td>$</td>
<td>$0.2e - 3$</td>
<td>$0.1e - 3$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_2)</td>
<td>$</td>
<td>$0.4e - 80$</td>
<td>$0.1e - 86$</td>
</tr>
<tr>
<td>$</td>
<td>f_1(x_3)</td>
<td>$</td>
<td>$0.5e - 1309$</td>
<td>$0.2e - 1412$</td>
</tr>
</tbody>
</table>
Solving (2.4) and simplifying (without the index \( n \) and using divided differences) yields in

\[
b_5 = \frac{f(z,x)WY(w-y) + Zf[w,x](y-z) + f[y,x](z-w)W - (w-y)(w-z)(y-z)f'(x)}{(YZ(z-y)f(w) + (w-z)((w-y)(y-z)f(x) + WZf(y)) + WY(y-w)f(z))},
\]

\[
b_2 = f'(x) + f(x)b_5,
\]

\[
b_4 = \frac{(z-x)f[y,x] + (x-y)f[z,x] + (y-z)b_2 + ((z-x)f(y) + (x-y)f(z))b_5}{(x-y)(x-z)(y-z)},
\]

\[
b_3 = f[w,x,x] + f[w,x]b_5 - (w-x)b_4,
\]

(2.5)

wherein \( y - x = Y, z - x = Z, w - x = W \). Now we have a powerful approximation of the first derivative of the function in the fourth step of (2.1), which doubles the convergence rate of the optimal 8th order methods. Therefore, we attain the following class, in which we have four evaluations of the function and one evaluation of the first-order derivative:

Any optimal 8th-order method = \[
\begin{cases}
  f(x_n), \text{ and } f'(x_n) \text{ are available}, \\
  f(y_n) \text{ is available}, \\
  f(z_n) \text{ is available},
\end{cases}
\]

(2.6)

\[
x_{n+1} = w_n - \frac{(1 + b_5(w_n - x_n)^2 f(w_n))}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3}.
\]

Now using any of the optimal three-point three-step methods, we could obtain a novel 16th order technique which satisfies the Kung-Traub hypothesis as well. Using the optimal 8th order method given by Wang and Liu in [11] ends in the following four-step method:

\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)},
\]

\[
z_n = x_n - \frac{f(x_n) f(x_n) - f(y_n)}{f'(x_n) f(x_n) - 2f(y_n)},
\]

\[
w_n = z_n - \frac{f(z_n)}{f'(x_n)} \left[ \frac{1}{2} + \frac{5f(x_n)^2 + 8f(x_n)f(y_n) + 2f(y_n)^2}{5f(x_n)^2 - 12f(x_n)f(y_n)} \left( \frac{1}{2} + \frac{f(z_n)}{f(y_n)} \right) \right],
\]

(2.7)

\[
x_{n+1} = w_n - \frac{(1 + b_5(w_n - x_n)^2 f(w_n))}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3}.
\]

**Theorem 2.1.** Assume that the scalar function \( f \) be sufficiently smooth in the real open domain \( I \subseteq \mathbb{R} \). Furthermore let \( e_n = x_n - r \) and \( c_k = (f^{(k)}(r))/(k!f'(r)) \), for \( k \in \mathbb{N} \), \( N \) is the set of natural numbers.
Then, the iterative method (2.7) has the optimal order of convergence 16 and satisfies the following error equation:

\[
e^{n+1} = \frac{1}{25c_3} \left[ c_3^3 \left( c_2^2 - c_3 \right)^2 \left( 11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4 \right) \right.
\]
\[
\times \left( 11c_2^4c_3 - 25c_2^2c_3^2 + 5c_2c_3c_4 + 5 \left( c_3^3 + c_2^2c_3 + c_3c_4 \right) \right] e^{16}_n + O(e^{17}_n).
\]  

(2.8)

Proof. Using Taylor series expansion around the simple root \( r \) in the \( n \)th iterate, [12], results in \( f(x_n) = f'(r)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + \cdots + O(e_n^{17})] \), and also \( f'(x_n) = f'(r)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + \cdots + O(e_n^{10})] \). Applying these expansions in the first step of (2.7), we have

\[
y_n - r = c_2e_n^2 + \left( -2c_2^2 + 2c_3 \right) e_n^3 + \left( 4c_2^3 - 7c_2c_3 + 3c_4 \right) e_n^4 + \cdots + O(e_n^{17}).
\]  

(2.9)

Note that to keep the prerequisites at the minimum, we only mention the simplified error equations at the end of first, second, and third steps. Because we also know that the first three steps are an eighth-order technique. Taylor series expanding in the second step of (2.7) by applying (2.9) yields in

\[
z_n - r = \left( c_3^3 - c_2c_3 \right) e_n^4 + 2 \left( 2c_2^4 - 4c_2^2c_3 + c_2^2 + c_2c_4 \right) e_n^5 + \cdots + O(e_n^{17}).
\]  

(2.10)

Using (2.10) and the third step of (2.7) gives us

\[
w_n - r = \frac{1}{5} c_2 \left( c_2^3 - c_3 \right) \left( 11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4 \right) e_n^8 + \cdots + O(e_n^{17}).
\]  

(2.11)

Now, we expand \( f(w_n) \), we obtain

\[
f(w_n) = \frac{1}{5} c_2 \left( c_2^3 - c_3 \right) \left( 11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4 \right) f'(r)e_n^8 + \cdots + O(e_n^{17}).
\]  

(2.12)

Subsequently, in the last step we have

\[
\frac{(1 + b_3(w_n - x_n))^2 f(w_n)}{f'(x_n) + 2b_3(w_n - x_n) + (3b_4 + b_3b_5)(w_n - x_n)^2 + 2b_4b_5(w_n - x_n)^3}
\]
\[
= \frac{1}{5} c_2 \left( c_2^3 - c_3 \right) \left( 11c_2^4 - 25c_2^2c_3 + 5c_3^2 + 5c_2c_4 \right) e_n^8 + \cdots + O(e_n^{17}).
\]  

(2.13)
Finally, using (2.12) and (2.13) in the last step of (2.7) ends in

\[ e_{n+1} = x_{n+1} - r = \frac{1}{25c_3} \left[ c_2^3 (c_2^2 - c_3)^2 \left( 11c_2^4 - 25c_2^2c_3 + 5c_2^2 + 5c_2c_4 \right) \right] \times \left( 11c_2c_3 - 25c_2^2c_3 + 5c_2c_3c_4 + 5\left( c_3^4 + c_4^2 - c_3c_5 \right) \right) e_n^{16} + O\left( e_n^{17} \right), \]

(2.14)

which shows that (2.7) is a 16th order method consuming only five evaluations per iteration. This completes the proof. \(\square\)

Remark 2.2. If we choose any of the other optimal 8th order derivative-involved methods in the first three steps of (2.6), then a novel optimal 16th order technique will be obtained. For instance, using the optimal 8th order method given by J. R. Sharma and R. Sharma in [13] results in the following four-step optimal method:

\[
\begin{align*}
\ y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\
\ z_n &= y_n - \frac{f(x_n) f(y_n)}{f(x_n) - 2 f(y_n) f'(x_n)}, \\
\ w_n &= z_n - \left[ 1 + \frac{f(z_n)}{f(x_n)} \right] \frac{f(x_n, y_n) f(z_n)}{f(y_n, z_n) f(x_n, z_n)}, \\
\ x_{n+1} &= w_n - \frac{(1 + b_3(w_n - x_n))^2 f(w_n)}{f'(x_n) + 2 b_3 (w_n - x_n) + (3b_4 + b_3 b_5)(w_n - x_n)^2 + 2 b_4 b_5 (w_n - x_n)^3},
\end{align*}
\]

(2.15)

which satisfies the following error equation:

\[
\begin{align*}
\ e_{n+1} &= \frac{c_1}{c_3} \left[ (c_2^2 - c_3)^2 \left( 3c_2^2 - 4c_2c_4 + c_4 \right) \left( 3c_2^4c_3 - 4c_2^2c_3^2 + c_2c_3c_4 + c_4^2 - c_3c_5 \right) \right] e_n^{16} + O\left( e_n^{17} \right).
\end{align*}
\]

(2.16)

Remark 2.3. Any method of the developed class carries out five evaluations per full cycle to reach the optimal order 16. Hence, the index of efficiency for our class is \(\sqrt[16]{16} \approx 1.741\), which is greater than that of optimal 4th order techniques \(\sqrt[4]{4} \approx 1.587\), and optimal 8th order techniques \(\sqrt[8]{8} \approx 1.682\).

3. Computational Experiments

The analytical outcomes given in the last section are fully supported by numerical experiments here. Two methods of the class (2.6), that is, methods (2.7) (S-S 16 (I)) and (2.15) (S-S 16 (II)), are compared with some of high-order methods.
For numerical comparisons, we have the 16th-order method of Geum and Kim [14] given as follows:

\[
y_n = x_n - f'(x_n)^{-1} f(x_n),
\]
\[
z_n = y_n - K_f f'(x_n)^{-1} f(y_n),
\]
\[
w_n = z_n - H_f f'(x_n)^{-1} f(z_n),
\]
\[
x_{n+1} = w_n - W_f f'(x_n)^{-1} f(w_n),
\]

wherein \( K_f = \frac{1 + 2u - 4u^2}{1 - 3u^2} \), \( H_f = \frac{1 + 2u}{1 - u} (1 - v + 2q) \), \( W_f = \frac{u(1 - 6u - u^2 - 24u^3)q + 2(1 - u)q^2 + (1 + 2u)(1 - t + v - 2q - 2vq)}{1 - t - v - 2q - 2vq} \), and \( u = f(y)/f(x), v = f(z)/f(y), q = f(z)/f(x), t = f(w)/f(z) \) (without the index \( n \)). This method is denoted by (G-K 16). We also used the 16th order method obtained by Neta and Petković (1.5) with \( t = 0 \) (N-P 16). We do not include other schemes with other orders, and one may refer to [15–21] for having further information about the other recent methods. Among many numerical examples, we have selected ten instances, with the use of multiprecision arithmetic. The test functions are displayed in Table 1.

For comparisons, 4000 digits floating points have been considered. As it shows, any method of the presented class is more efficient than the existing high order methods in the literature. Results in Table 2 manifest the applicability of the new scheme in the test problems. Also notice that F. stands for failure, that is, the starting value is not sufficiently close to the zero to make the iterative method converge.

We have checked that the sequence of iterations converge to an approximation of the solution for nonlinear equations in our numerical works. We note that an important problem of determining good starting points appears, when applying iterative methods for solving nonlinear equations. A quick convergence, one of the advantages of multipoint methods, can be attained only if initial approximations are sufficiently close to the sought roots; otherwise, it is not possible to realize the expected convergence speed in practice. For achieving such a goal, we remind the well-known technique of interval Newton’s method as a tool to obtain robust initial approximations in the next section.

### 4. All the Zeros

In this section, we pay close attention to the matter of finding all the (simple) zeros of nonlinear functions in an interval using a hybrid algorithm. In fact, in practice one wish to find all the zeros at the same time. On the other hand, the iterative scheme, such as (2.7), is so sensitive upon the choice of the initial guess. Generally speaking, no method is the best iterative scheme in all test problems. Mostly, this is due to the test function and the accuracy of the initial approximation.

Having a sharp enough initial estimation, one may start the process as efficiently as possible and to see the computational order of convergence correctly, that is, the rate of correcting the number decimal places to the true solution. However, having such an initial guess is not always an easy task for practical problems.

In what follows, we remind a well-known scheme based on [22], also known as interval Newton’s method for finding enough accurate initial approximations for all the zeros.
of a nonlinear function in an interval. Hopefully, Mathematica 8 gives the users a friendly environment for working with lists (sequences), that is to say, the obtained list then would be corrected using the new 16th order schemes of this paper.

This procedure was efficiently coded in [22] and with some changes we have provided in what follows:

```mathematica
intervalnewton::rec = "MaxRecursion exceeded.";
intnewt[f_, df_, x_, {a_, b_}, eps_, n_] :=
    Block[{{xmid, int = Interval[{a, b}]}, If[b - a < eps, Return[int]]; If[n == 0, Message[intervalnewton::rec]; Return[int]; xmid = Interval[SetAccuracy[(a + b)/2, 16]]; int = IntervalIntersection[int, SetAccuracy[xmid - N[f/.x->xmid]/N[df/.x->int], 16]]; (intnewt[f, df, x, #, eps, n - 1]) & /@ (List @@ int)]; Options[intervalnewton] = {MaxRecursion -> 2000};
```

The above piece of code takes the nonlinear function \( f \), the lower and upper bounds of the working interval and applies the interval form of the Newton's iteration with the maximum number of recursion considered as 2000 and also the use of the command `SetAccuracy[exp, 16]`. Note that in this case, like the Newton's method, the function should have first order derivative on the interval. Thus now, the call-out function to implement the above piece of code could be given by

```mathematica
intervalnewton[f_, x_, int_, eps_, opts__] := Block[{{df, n}, n = MaxRecursion /. {opts} /. Options[intervalnewton]; df = D[f, x]; IntervalUnion @@ Select[Flatten[(intnewt[f, df, x, #, eps, n]) & /@ (List @@ int)], IntervalMemberQ[f/.x->#, 0] &];
```

The tolerance should be chosen to be arbitrary in machine precision. To illustrate the procedure further, we consider the oscillatory nonlinear function \( f(x) = e^{\sin(\log(x)\cos(20x))} - 2 \), where its plot has been given in Figure 1, in the interval \([a, b] = [2, 10.]\) while the tolerance is set to 0.0001 as follows:

```mathematica
f[x_] := Exp[Sin[Log[x]*Cos[20x]]] - 2;
IntervalSol = intervalnewton[f[x], x, Interval[{2., 10.}], .0001];
setInitial = N[Mean@List@IntervalSol]; NumberOfGuesses = Length[setInitial];
```

5. Conclusions

A class of four-point four-step iterative methods has been developed for solving nonlinear equations. The analytical proof for one method of the class was written to clarify the 16th order convergence. The class was attained by approximating the first derivative of the function in the fourth step of a cycle in which the first three steps are any of the optimal eighth-order derivative-involved methods.

Per full cycle, the methods of the class consist of four evaluations of the function and one evaluation of the first derivative which results in 1.741 as the efficiency index. The presented class satisfies the still unproved conjecture of Kung-Traub on multi-point iterations without memory.

The accuracy and efficiency of two obtained methods of the class were illustrated by solving a lot of numerical examples. Numerical works also have attested the theoretical results given in the paper and have shown the fast rate of convergence.

Acknowledgment

The authors cheerfully thank the remarks of the two reviewers on an earlier version of this paper.

References


